

達人プログラマーの感想

安彦 久志

2021/02/17

1 はじめに

ソフトウェア開発全般に関わる書籍は初めて読んだ。研究のために行っていたソフトウェア開発と本書を比較して、特に気になった内容について述べる。

2 割れた窓を放置しておかないこと

完全に放置することはなかった。対処としては、コメントを残す。バグが疑われる箇所や修正すべき箇所をリストアップしておく。などをしていた。

3 DRY—繰り返しを避けること

本書では二重化を4つのカテゴリに分割しており、それぞれ思い当たる節があった。なかでも最も多くやってしまったことは「手抜きによる二重化」であった。例えば、3回程度の同じ処理であればfor文を使わずに、記述することがよくあった。for文を使うよりも、コピー&ペーストの方がタイプする回数が少ないためだった。パラメータの設定を2つのプログラムでコードに直書きしていることがあった。その数値は変更しない前提で開発していたが、数値に誤りがあり、修正するとき一方のコードしか修正しておらず、期限ギリギリになってバグが見つかり、この二重化に気づくのには時間が掛かったことがあった。

4 モジュール感の結合度を最小にする

当初は、C++で開発を試みたが、クラス設計をどうして良いかわからず、早期にC言語に切り替えた。しかし、開発が進むと関数に複数の構造体を渡すことが多発した。また、主要な構造体（顧客、タクシー、道路網）はグローバル変数で定義しており、煩雑なコードになってしまった。

5 テスト設計を行うこと

開発の中でテストは注意深く行っており、多くのテストをコード内に実装していた。これにより、多くのバグを発見することができ、かなり役立った。しかし、全ての機能をテストすることは出来ておらず、テストコードを組み込む箇所は感覚に頼っていた。これに対して、本書ではコードの実装より前にテストを作成することを考えている。期限に追われるなかでテストを先に作成することは、現実的ではないようにも感じた。しかし、テストを作成することは、ソフトウェアの出力について、制約条件を確認することになるので、システムを俯瞰して考えることができると感じた。

6 ドキュメントは付け足すものでなく、組み込むものである

これまでドキュメントを書いたことはなく、コーディングとは別の作業だと想像していた。しかし、コード内に適切なコメントを記述し、コメントからドキュメントを作成するようなツールを用いることで、容易にドキュメントを作成できることは効率的だと感じた。これにより、ドキュメントはコードに対する単なるビューとなるという考えは、プログラマらしく面白いと思った。

7 おわりに

ほどほどに難しくて、応用範囲の広い本は数年読み続けられると感じた。例えば「柳浦睦憲・茨木俊秀，“組合せ最適化—メタ戦略を中心として—”，2001年，朝倉書店」など。